UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/608,512	06/30/2000	Reynold V. D'Sa	2207/P6786	9566

7590 11/05/2004

Kenyon & Kenyon
333 W San Carlos Street Suite 600
San Jose, CA 95110

EXAMINER	
HARKNESS, CHARLES A	
ART UNIT	**PAPER NUMBER**
2183	

DATE MAILED: 11/05/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	**Application No.** 09/608,512	**Applicant(s)** D'SA ET AL.	
	Examiner Charles A Harkness	**Art Unit** 2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *18 August 2004*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-26* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-26* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application (PTO-152)

6) ☐ Other: _____.

DETAILED ACTION

Claim Rejections - 35 USC § 103

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

2. Claims 7-13 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kahle et al, U.S. Patent Number 5,956,495 (herein referred to as Kahle) in view of McCrocklin et al, U.S. Patent Number 4,761,733 (herein referred to as McCrocklin).

3. Referring to claim 7 Kahle has taught a method of detecting bogus branch instructions in a microprocessor instruction pipeline, the method comprising:

predicting whether a first instruction is a bogus branch instruction (Kahle column 9 lines 56-65 column 10 lines 49-56); and

looking "ahead" in the instruction pipeline to at least one second instruction related to the first instruction, wherein if the first micro-op is predicted to be a bogus branch, the method further comprises:

attaching a signal flag that indicates a bogus branch to the at least one second instruction (Kahle column 10 lines 56-67 column 9 lines 56-65 column 11 lines 1-35).

Kahle has not taught using microinstructions. It would have been obvious to one of ordinary skill in the art at the time of the invention to use microinstruction processors because modern computer systems typically include a micro-programmable microprocessor, which will utilize macroinstructions and microinstructions (McCrocklin column 1 lines 9-33). Since this is a typical embodiment for microprocessors, as shown by McCrocklin, one of ordinary skill in the art would have recognized the benefit in using microinstructions and macroinstructions, where the tasks are broken down into more basic functions which can be executed faster than one complex instruction. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to use a microprocessor with microinstructions to speed up the execution of the instructions.

4. Referring to claim 8 McCrocklin has taught further comprising: decoding at least one macro instruction into the first micro-op and the at least one second micro-op (McCrocklin column 1 lines 23-33); and writing the first micro-op and the at least one second micro-op into a decoded micro-op cache (McCrocklin column 6 lines 49-61, column 7 lines 25-49).

5. Referring to claim 9 Kahle has taught wherein the prediction of whether the first micro-op is a bogus branch instruction is based on branch prediction logic (Kahle column 9 lines 56-65 column 10 lines 49-56).

6. Referring to claim 10 Kahle has taught a method of recovering from a bogus branch instruction in a microprocessor instruction pipeline, the method comprising: determining whether a first instruction is a bogus branch (Kahle column 10 lines 56-67 column 9 lines 56-65 column 11 lines 1-35)

Deallocating from a decoded instruction cache at least one second instruction related to the first instruction (Kahle column 10 lines 60-67, column 8 lines 36-40 column 11 lines 18-25).

Kahle has not taught using microinstructions. McCrocklin has taught using microinstructions. It would have been obvious to one of ordinary skill in the art at the time of the invention to use microinstruction processors because modern computer systems typically include a micro-programmable microprocessor, which will utilize macroinstructions and microinstructions (McCrocklin column 1 lines 9-33). Since this is a typical embodiment for microprocessors, as shown by McCrocklin, one of ordinary skill in the art would have recognized the benefit in using microinstructions and macroinstructions, where the tasks are broken down into more basic functions which can be executed faster than one complex instruction. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to use a microprocessor with microinstructions to speed up the execution of the instructions.

7. Referring to claim 11 Kahle has taught wherein determining whether the first micro-op is a bogus branch is based on branch prediction logic (Kahle column 9 lines 56-65 column 10 lines 49-56).

8. Referring to claim 12 Kahle has taught wherein deallocating from the decoded micro-op cache the at least one second micro-op is accomplished by checking whether a bogus branch signal "flag" has been attached to the at least one second micro-op (Kahle column 10 lines 60-67, column 8 lines 36-40 column 11 lines 18-25).

9. Referring to claim 13 Kahle has taught wherein deallocating further comprises at least one of:

removing the specific bogus branch;

removing all branches in a set with the bogus branch;

removing all branches in the decoded micro-op cache; and

clearing the entire decoded micro-op cache (Kahle column 10 lines 60-67, column 8 lines 36-40 column 11 lines 18-25).

10. Claims 1-6 and 14-26 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kahle in view of McCrocklin in view of Shiell U.S. Patent 5,864,697 (herein referred to as Shiell).

11. Referring to claim 1 Kahle has taught a method of detecting, recovering from and preventing bogus branch instructions in a microprocessor, the method comprising:

predicting by branch prediction logic whether the at least one micro-op is a branch (Kahle column 9 lines 56-65 column 10 lines 49-56);

executing the at least one micro-op (Kahle column 11 lines 1-35; the execution units execute all instructions, including branches; the branch instruction has to be executed to determine if the prediction was correct column 5 lines 35-43);

determining if the at least one executed micro-op is a bogus branch of the first macro instruction (Kahle column 10 lines 56-67 column 9 lines 56-65 column 11 lines 1-35); and

continuing processing with a second macro instruction (it is inherent that the system will continue to execute instructions, regardless of which path the branch takes, presuming that the path does not lead to the end of the program, otherwise the program would not finish executing)

wherein if the at least one executed micro-op is determined to be branch, then the method further comprises:

flagging any other micro-ops which pertain to the at least one executed bogus branch micro-op (column 10 lines 60-67);

removing the flagged micro-ops for retirement (column 8 lines 36-40 column 11 lines 18-25).

Kahle has not taught:

decoding a first macro instruction into at least one micro-op;

writing the at least one micro-op into a decoded micro-op cache; and

scrubbing a branch prediction logic storage buffer upon which the branch prediction logic is based (would have been obvious)

McCrocklin has taught:

decoding a first macro instruction into at least one micro-op (McCrocklin column 1 lines 23-33);

writing the at least one micro-op into a decoded micro-op cache (McCrocklin column 6 lines 49-61, column 7 lines 25-49);

It would have been obvious to one of ordinary skill in the art at the time of the invention to use microinstruction processors because modern computer systems typically include a micro-programmable microprocessor, which will utilize macroinstructions and microinstructions (McCrocklin column 1 lines 9-33). Since this is a typical embodiment for microprocessors, as shown by McCrocklin, one of ordinary skill in the art would have recognized the benefit in using microinstructions and macroinstructions, where the tasks are broken down into more basic functions which can be executed faster than one complex instruction. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to use a microprocessor with microinstructions to speed up the execution of the instructions.

Kahle and McCrocklin have not taught scrubbing a branch prediction logic storage buffer upon which the branch prediction logic is based. This would be an improvement that one of ordinary skill in the art at the time of the invention would include with the system of Kahle, because without such a function, the prediction logic will never change and will not learn from branch history, allowing the branch prediction logic to make better predictions. This is shown in Shiell where the older positions in the branch history table are invalidated (Shiell column 14 lines 27-44). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to update the branch prediction logic, or deallocate the old, or unneeded, entries so the prediction logic takes advantage of the branch history by predicting branch instructions based on previous instructions, allowing the prediction to be more accurate and speeding up the time of execution.

12. Referring to claim 2 the combination of Kahle and McCrocklin has taught further comprising: fetching from a main memory the macro instruction (McCrocklin column 1 lines 23-33).

13. Referring to claim 3 Kahle has taught wherein the at least one micro-op is written into the decoded micro-op cache in an order a branch table buffer predicts that the at least one micro-op should be executed (Kahle column 2 lines 37-40, abstract).

14. Referring to claim 4 Kahle has taught wherein executing the at least one micro-op is in at least one of an "in-order" or "out-of-order" fashion (Kahle column 2 lines 37-40, abstract).

15. Referring to claim 5 the combination of Kahle, McCrocklin, and Shiell has taught wherein scrubbing the branch prediction logic storage buffer further comprises at least one of:

deallocating any other micro-ops pertaining to the at least one executed bogus branch micro-op;

deallocating at least one old set which had been overwritten in the decoded micro-op cache by a built instruction "trace";

deallocating at least one entry that is related to a branch in at least one old set in the decoded micro-op cache; and

deallocating at least one entry that is related to a branch of at least one old set in the decoded micro-op cache that is downstream from the at least one executed bogus branch micro-op (Shiell column 14 lines 27-44; Shiell has taught invalidating positions, or deallocating at least one entry that is related to a branch in an old set).

16. Referring to claim 6 Kahle has taught further comprising: determining if the branch has been taken (Kahle column 10 lines 56-67 column 9 lines 56-65 column 11 lines 1-35).

17. Referring to claim 14 Kahle has taught a method of preventing a bogus branch instruction from being executed in a microprocessor instruction pipeline (Kahle column 9 lines 56-65 column 10 lines 49-56), the method comprising: retiring the at least one instruction (Kahle column 7 line 60-column 8 line 8 column 8 lines 36-40 column 11 lines 18-25).

Kahle has not taught using microinstructions and writing at least one micro-op into a decoded micro-op. McCrocklin has taught using microinstructions and writing at least one micro-op into a decoded micro-op cache (McCrocklin column 6 lines 49-61, column 7 lines 25-49).

It would have been obvious to one of ordinary skill in the art at the time of the invention to use microinstruction processors because modern computer systems typically include a micro-programmable microprocessor, which will utilize macroinstructions and microinstructions (McCrocklin column 1 lines 9-33). Since this is a typical embodiment for microprocessors, as shown by McCrocklin, one of ordinary skill in the art would have recognized the benefit in using microinstructions and macroinstructions, where the tasks are broken down into more basic functions which can be executed faster than one complex instruction. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to use a microprocessor with microinstructions to speed up the execution of the instructions. Kahle and McCrocklin have not taught scrubbing a branch prediction logic storage buffer upon which the branch prediction logic is based. This would be an improvement that one of ordinary skill in the art at the time of the invention would include with the system of Kahle, because without such a function, the prediction logic will never change and will not learn from branch history, allowing the branch prediction logic to make better predictions. This is shown in Shiell where the older positions in the branch history table are invalidated (Shiell column 14 lines 27-44). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to update the branch prediction logic, or deallocate the old, or unneeded, entries so the prediction logic takes advantage of the branch history by predicting branch instructions based on previous instructions, allowing the prediction to be more accurate and speeding up the time of execution.

18. Referring to claim 15 Kahle has taught wherein retiring the at least one micro-op comprises at least:

determining what the actual result for the retired at least one micro-op was (Kahle column 10 lines 56-67 column 9 lines 56-65 column 11 lines 1-35, column 7 line 60-column 8 line 8; the result of the branch is determined after a prediction is made when the branch instruction is executed; column 8 lines 36-40 column 11 lines 18-25).

19. Referring to claim 16 Shiell has taught wherein scrubbing the branch prediction logic storage buffer comprises at least:

comparing what an actual result of the retired at least one micro-op is to an instruction trace in the branch prediction logic storage buffer (Shiell column 2 lines 11-31,abstract).

20. Referring to claim 17 the combination of Kahle, McCrocklin, and Shiell has taught wherein scrubbing the branch prediction logic storage buffer further comprises at least one of:

deallocating any other micro-ops pertaining to the at least one retired micro-op;

deallocating at least one old set which had been overwritten in the decoded micro-op cache by a built instruction "trace";

deallocating at least one entry that is related to a branch in at least one old set in the decoded micro-op cache; and

deallocating at least one entry that is related to a branch of at least one old set in the decoded micro-op cache that is downstream from the at least one retired micro-op (Shiell column 14 lines 27-44; Shiell has taught invalidating positions, or deallocating at least one entry that is related to a branch in an old set).

21. Referring to claim 18 the combination of Kahle, McCrocklin, and Shiell has not taught wherein scrubbing can be accomplished at the time of at least one of writing or retiring. Official Notice is taken that reducing the time of the scrubbing of the prediction logic to the same time that it takes to retire the instruction or write the instruction would benefit the system because the system will not have to wait on the scrubbing of the prediction logic before continuing with the execution of the program, and that reducing the amount of time that any process takes in general will benefit the system by allowing the program to be executed in less time.

22. Referring to claim 19 Kahle has taught an apparatus for detecting; recovering and preventing bogus branch instructions in a microprocessor, the method comprising:

a branch prediction logic storage buffer for predicting whether a branch will be taken upon execution of the at least one decoded micro-op (Kahle column 10 lines 49-67);

an instruction execution unit for executing the at least one micro-op (Kahle column 11 lines 1-35; the execution units execute all instructions, including branches; the branch instruction has to be executed to determine if the prediction was correct column 5 lines 35-43); and

an instruction retirement unit which determines whether the at least one micro-op is of a bogus branch macro instruction, wherein if the instruction retirement unit determines the at least one micro-op is of a bogus branch macro instruction (Kahle column 10 lines 60-67 column 10 lines 56-67 column 9 lines 56-65 column 11 lines 1-35, column 7 line 60-column 8 line 8; the result of the branch is determined after a prediction is made when the branch instruction is executed; column 8 lines 36-40 column 11 lines 18-25);

any other micro-ops stored in the decoded micro-op cache pertaining to that bogus branch macro instruction are flagged (Kahle column 10 lines 60-67) and

removed to the instruction retirement unit for retirement (Kahle column 10 lines 56-67 column 9 lines 56-65 column 11 lines 1-35, column 7 line 60-column 8 line 8; the result of the branch is determined after a prediction is made when the branch instruction is executed; column 8 lines 36-40 column 11 lines 18-25).

Kahle has not taught a decoded micro-op cache into which are written at least one decoded micro-op of a macro instruction. McCrocklin has taught a decoded micro-op cache into which are written at least one decoded micro-op of a macro instruction (McCrocklin column 1 lines 23-33, column 6 lines 49-61, column 7 lines 25-49). It would have been obvious to one of ordinary skill in the art at the time of the invention to use microinstruction processors because modern computer systems typically include a micro-programmable microprocessor, which will utilize macroinstructions and microinstructions (McCrocklin column 1 lines 9-33). Since this is a typical embodiment for microprocessors, as shown by McCrocklin, one of ordinary skill in the art would have recognized the benefit in using microinstructions and macroinstructions, where the tasks are broken down into more basic functions which can be executed faster than one complex instruction. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to use a microprocessor with microinstructions to speed up the execution of the instructions.

Kahle and McCrocklin have not taught scrubbing a branch prediction logic storage buffer upon which the branch prediction logic is based. This would be an improvement that one of ordinary skill in the art at the time of the invention would include with the system of Kahle, because without such a function, the prediction logic will never change and will not learn from branch history, allowing the branch prediction logic to make better predictions. This is shown in Shiell where the older positions in the branch history table are invalidated (Shiell column 14 lines 27-44). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to update the branch prediction logic, or deallocate the old, or unneeded, entries so the prediction logic takes advantage of the branch history by predicting branch instructions based on previous instructions, allowing the prediction to be more accurate and speeding up the time of execution.

23. Referring to claim 20 the combination of Kahle and McCrocklin has taught further comprising: a main memory in which the macro instruction is stored; and an instruction fetch

unit for fetching the macro instruction from the main memory (McCrocklin column 1 lines 23-33).

24. Referring to claim 21 McCrocklin has taught further comprising: an instruction decode unit for translating the macro instruction into the at least one decoded micro-op (McCrocklin column 1 lines 23-33; it is inherent that if the system decodes the macroinstructions into microinstructions that some decode unit must exist in the system).

25. Referring to claim 22 Kahle has taught further comprising: a jump execution unit which determines whether a branch was taken upon execution of the at least one decoded micro-op (Kahle column 10 lines 56-67 column 9 lines 56-65 column 11 lines 1-35).

26. Referring to claim 23 the combination of Kahle, McCrocklin, and Shiell has taught wherein the branch prediction logic storage buffer applies branch prediction logic to predict whether a branch will be taken upon execution of the at least one decoded micro-op (Kahle column 9 lines 56-65 column 10 lines 49-56).

27. Referring to claim 24 the combination of Kahle, McCrocklin, and Shiell has taught wherein if the branch prediction logic storage buffer predicts a branch will be taken upon execution of the at least one decoded micro-op, an instruction "trace" is built pertaining to the predicted branch (Kahle column 10 lines 49-67).

28. Referring to claim 25 the combination of Kahle, McCrocklin, and Shiell has taught wherein the built instruction "trace" is inserted into the decoded micro-op cache such that the micro-ops of the branch macro-instruction are executed (Kahle column 9 lines 56-65 column 10 lines 49-56; with the combination of Kahle and McCrocklin, the trace, or flags that Kahle uses to

flag the instructions associated with a branch would be put on to the instruction still awaiting to be executed in the cache after being fetched based on the branch prediction).

29. Referring to claim 26 the combination of Kahle, McCrocklin, and Shiell has taught wherein the branch prediction logic storage buffer is scrubbed by deallocation of at least one of any other micro-ops pertaining to the bogus branch macro instruction, any old set which had been overwritten in the decoded micro-op cache by a built instruction "trace", all entries that are related to any branches in the old set, and all entries that are related to the branches in the old set that are downstream from the retired branch macro instruction (Shiell column 14 lines 27-44; Shiell has taught invalidating positions, or deallocating at least one entry that is related to a branch in an old set).

30. The rejections of claims 1-26 have been maintained.

Response to Arguments

31. Applicant's arguments filed 08/18/04 have been fully considered but they are not persuasive.

32. In the remarks, in regard to the rejection of claims, Applicant argues in essence that:

> "In other words, Kahle describes determining whether of a branch has been predicted to be taken or not taken, as opposed to determining if a branch instruction is bogus, as required by claims 7 and 10. For example, a bogus branch is described as a branch that is predicted to occur at an address that does not contain a branch or will have a target address that is invalid. Therefore claims 7 and 10 are not obvious under Kahle in view of McCrocklin."

33. This is not found persuasive. Applicant has not provided an explicit definition of a "bogus branch", either in the claims or in the specification. In the remarks on page 11, on line 4 , Applicant states, "For example, a bogus branch is described as..." and in the specification on

page 2 line 12 Applicant states, "Such 'bogus branch' predications..." Exemplification is not an explicit definition and cannot be read into the claims. Since there is no explicit definition by Applicant, the broadest reasonable interpretation is being used by the Examiner during prosecution of the instant application. See MPEP 2111.01. Examiner has interpreted "bogus branch" to mean a mispredicted branch instruction, or where a branch instruction is predicted to be taken or not taken, and the prediction is false.

Kahle has clearly shown that recovery will take place if a misprediction occurs (Kahle column 10 49-67). Therefore, Kahle must be able to detect when a branch does not act as predicted, or is a bogus branch.

34. In the remarks, in regard to the rejection of claims, Applicant argues in essence that:

> "Applicants respectfully submit that neither Kahle, McCrocklin, Shiell, nor any combination thereof disclose removing entries from a branch prediction logic storage buffer that may later produce bogus branches, as recited in claim 14."

35. This is not found persuasive. The language "may" does not limit the claim 14 since the method being taught has taught entries that "may" later produce bogus branches. Since the entries "may" or "may not" later produce bogus branches, the claim is not limited with the language shown.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Charles A Harkness whose telephone number is 703-305-7579 and 571-272-4167 after 10/12/04. The examiner can normally be reached on 8Flex.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on 517-272-4162. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Charles Allen Harkness

Examiner

Art Unit 2183

October 28, 2004

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100